

A game theoretic approach to a network cloud storage problem

Fabio Fagnani*

Barbara Franci†

January 23, 2017

Abstract

The use of game theory in the design and control of large scale networked systems is becoming increasingly more important. In this paper, we follow this approach to efficiently solve a network allocation problem motivated by peer-to-peer cloud storage models as alternatives to classical centralized cloud storage services. To this aim, we propose an allocation algorithm that allows the units to use their neighbors to store a back up of their data. We prove convergence, characterize the final allocation, and corroborate our analysis with extensive numerical simulation that shows the good performance of the algorithm in terms of scalability, complexity and structure of the solution.

Key words. Network Allocation Problem, Cloud storage, Evolutionary Game Theory, Noisy Best Response, Systems on Graphs

AMS subject classification. 91A22, 91A43, 90C27

1 Introduction

In the last decade, game theory has emerged as a new fundamental paradigm in the design and control of large scale networked engineering systems [2, 4, 6]. The basic idea is that of modeling system units as rational agents whose behavior consists in a selfish search for their maximum utility. The goal is to design both the agents utility functions and an adaptation rule (evolutionary game dynamics) in such a way that the resulting global behavior is desirable. One of the advantages of this approach with respect to classical global optimization techniques is that it naturally leads to scalable decentralized solutions that can easily incorporate constraints on the computational power of the units as well as the information flow determined by the network architecture. Consensus dynamics, which is a fundamental ingredient of networked control, has been recently recognized [5] to be an instance of an evolutionary dynamics of a networked coordination game.

Two are the challenging issues when (evolutionary) game theory is used as a mechanism design: first, designing admissible agent utility functions such that the resulting game possesses desirable Nash equilibria; second, designing adaptation rules that guarantee the solution's convergence to such Nash equilibria. For an important class of games known as potential games, a popular adaptation rule is the so-called noisy best-response where units choose their state by following a Gibbs-Boltzmann distribution having its peak on the maxima of the utility function [4].

*Department of Mathematical Sciences, Politecnico di Torino, Italy (fabio.fagnani@polito.it).

†Department of Mathematical Sciences, Politecnico di Torino, Italy (barbara.franci@polito.it).

This rule can be proven to yield convergence, asymptotically in time and in the limit of the noise approaching zero, to maxima of the potential that form a subclass of the Nash equilibria. In this context, the goal is to design utility functions in such a way that the potential (in particular, its maxima) have the desired global properties. The noisy best-response turns out to be a random decentralized scheme for the maximization of the potential.

In this paper, we consider an allocation problem where a population of units is connected through a network and each of them possesses a number of items that need to be allocated among the neighboring units. The interpretation is that of a decentralized peer-to-peer cloud storage model where computers (or other devices with storage capabilities) in a network are made to collaborate with each other so that each of them can store a back up of their data in the available space of the neighbors. The model was first presented in [1] where the reader can find more detailed motivations.

We cast the allocation problem into a game theoretic framework by assuming that each unit has a utility function that gives a value to their neighbors on the basis of their reliability (possibly reflecting responsiveness or security issues), their current congestion (resources have bounded storage capabilities), and the amount of data the unit already stores in them. This reflects a number of features that we want the storage final configuration to possess. In the storage process, a unit should prefer to use those neighbors that are more reliable and, among equally reliable ones, those that are less congested. Finally, storage should not be split into too many parts to avoid excessive communication burden. Following evolutionary game theory [4], we propose an algorithm based on a noisy best response action: each time a unit activates, it decides which neighbor to use for storage on the basis of a Gibbs probability distribution.

Allocation games have been extensively studied [8, 3]. One of the key features of our model is the presence of hard constraints as a consequence of the bounded storage capabilities of the units. This property is non-classical in game theory and has remarkable consequences on the structure of Nash equilibria and the behavior of the algorithm. The proposed game is shown to be a potential game. However, convergence of the best response dynamics is here a subtle issue that does not follow from classical results. This will follow instead from a careful proof of the ergodicity of the underlying Markov process that constitutes the technical core of this paper. The conclusive result is that: (i) all units complete their allocation in finite time with probability one; and (ii) the allocation configuration converges, when the noise parameter approaches 0, to a maximum of the potential. These results were announced in [1], but not proven. Moreover, in this paper, we deal with a more general situation: units can be in off mode and hence not responsive to storage requests from neighbors.

In the remaining part of this section, we formally define the storage allocation problem and recall some basic facts proven in [1] (in particular, a necessary and sufficient condition for the allocation problem to be solvable). Section 2 is devoted to cast the problem to a potential game theoretic framework [7] and to propose a distributed algorithm that is an instance of a noisy best response dynamics. The main technical part of the paper is Section 3 where the fundamental results Theorem 3 and Corollary 8 are stated and proven. Theorem 3 ensures that the algorithm reaches a complete allocation with probability one, if a complete allocation is indeed possible. Corollary 8 studies the asymptotic behavior of the algorithm and explicitly

exhibits the invariant probability distribution. Consequence of Corollary 8 is that in the double limit when time goes to infinity and the noise parameter goes to 0, the algorithm converges to a Nash equilibrium that is, in particular, a global maximum of the potential function. This guarantees that the solution will indeed be close to the global welfare of the community. Finally, Section 4 is devoted to the presentation of an extensive set of simulations that corroborate the theoretical results, prove the good scalability properties of the algorithm in terms of speed and complexity, and illustrate the effect of the parameters of the utility functions in the solution reached by the algorithm. A conclusion section ends the paper.

1.1 The model

Consider a set \mathcal{X} of units that play the double role of users who have to allocate externally a back up of their data, as well resources where data from other units can be allocated. Generically, an element of \mathcal{X} will be called a unit, while the terms user and resource will be used when the unit is considered in the two possible roles of, respectively, a source or a recipient of data. We assume units to be connected through a directed graph $\mathcal{G} = (\mathcal{X}, \mathcal{E})$ where a link $(x, y) \in \mathcal{E}$ means that unit x is allowed to store data in unit y . We denote by

$$N_x := \{y \in \mathcal{X} \mid (x, y) \in \mathcal{E}\}, \quad N_y^- := \{x \in \mathcal{X} \mid (x, y) \in \mathcal{E}\}$$

respectively, the out- and the in-neighborhood of a node. Note the important different interpretation in our context: N_x represents the set of resources available to unit x while N_y^- is the set of units having access to resource y . If $D \subseteq \mathcal{X}$, we put $N(D) = \cup_{x \in D} N_x$ and $N^-(D) = \cup_{x \in D} N_x^-$.

We imagine the data possessed by the units to be quantized atoms of the same size. Each unit x is characterized by two non negative integers:

- α_x is the number of data atoms that unit x needs to back up into his neighbors,
- β_x is the number of data atoms that unit x can accept and store from his neighbors.

The numbers $\{\alpha_x\}$ and $\{\beta_x\}$ will be assembled into two vectors denoted, respectively, α and β . We also define

$$\mathcal{A}_x = \{(x, a) \mid a \in \{1, \dots, \alpha_x\}\}, \quad \mathcal{A} = \bigcup_{x \in \mathcal{X}} \mathcal{A}_x$$

Given the triple $(\mathcal{G}, \alpha, \beta)$, we define an allocation as any map $Q : \mathcal{A} \rightarrow \mathcal{X}$ ($Q(x, a) = y$ means that x has allocated the data atom a into resource y) satisfying the properties expressed below.

(C1) **Graph constraint** $Q(x, a) \in N_x$ for all $x \in \mathcal{X}$ and $a \in \{1, \dots, \alpha_x\}$;

(C2) **Storage limitation** For every $y \in \mathcal{X}$, it holds, $|Q^{-1}(y)| \leq \beta_y$.

We will say that the allocation problem is solvable if an allocation Q exists. We denote by \mathcal{Q} the set of allocations. We will also need to consider partial allocations, namely functions $Q : D \rightarrow \mathcal{X}$ where $D \subseteq \mathcal{A}$ and satisfying, where defined, conditions (C1) and (C2). We denote by \mathcal{Q}_p the set of partial allocations.

The following result gives a necessary and sufficient condition for the existence of allocations. The proof, which follows from Hall's theorem, can be found in [1].

Theorem 1. *Given $(\mathcal{G}, \alpha, \beta)$, there exists an allocation iff the following condition is satisfied:*

$$\sum_{x \in D} \alpha_x \leq \sum_{y \in N(D)} \beta_y \quad \forall D \subseteq \mathcal{X} \quad (1)$$

In general, it is not necessary to check the validity of (1) for every subset D . We say that $D \subseteq \mathcal{X}$ is maximal if for any $D' \supsetneq D$, it holds $N(D') \not\supseteq N(D)$. We say that $D_1, D_2 \subseteq \mathcal{X}$ are independent if $N(D_1) \cap N(D_2) = \emptyset$ and $D \subseteq \mathcal{X}$ is called irreducible if it can not be decomposed into the union of two non empty independent subsets. Clearly, it is sufficient to verify (1) for the subclass of maximal irreducible subsets.

Example 1. *If \mathcal{G} is complete, we have that $N(\{x\}) = \mathcal{X} \setminus \{x\}$ while $N(D) = \mathcal{X}$ for all D such that $|D| \geq 2$. Hence, the only maximal irreducible subsets are the singletons $\{x\}$ and the set \mathcal{X} . Condition (1) in this case reduces to*

$$\alpha_x \leq \sum_{y \neq x} \beta_y, \quad \forall x \in \mathcal{X} \quad \sum_{x \in \mathcal{X}} \alpha_x \leq \sum_{y \in \mathcal{X}} \beta_y \quad (2)$$

In general, the class of maximal irreducible subsets can be large and grow more than linearly in the size of \mathcal{X} , as the following example shows.

Example 2. *If $\mathcal{G} = (\mathcal{X}, \mathcal{E})$ is a line graph ($\mathcal{X} = \{1, 2, \dots, n\}$ and $\mathcal{E} = \{(i, i+1), i = 1, \dots, n-1\}$) it can be checked that the maximal irreducible subsets are those of the form $\{i, i+2, \dots, i+2s\}$.*

When the allocation problem is solvable, it will in general admit many possible solutions. In order to analyze the set of solutions it is convenient to introduce an aggregative function of allocations. Given a partial allocation Q , consider the matrix $W(Q) \in \mathbb{N}^{\mathcal{X} \times \mathcal{X}}$ where $W(Q)_{xy}$ is the number of atomic data that x has allocated in y under Q , namely,

$$W(Q)_{xy} := |Q^{-1}(y) \cap \mathcal{A}_x| \quad (3)$$

Clearly $W = W(Q)$ satisfies the following conditions

(P1) $W_{xy} \geq 0$ for all x, y and $W_{xy} = 0$ whenever $(x, y) \notin \mathcal{E}$.

(P2) $W_x := \sum_{y \in \mathcal{X}} W_{xy} \leq \alpha_x$ for all $x \in \mathcal{X}$.

(P3) $W^y := \sum_{x \in \mathcal{X}} W_{xy} \leq \beta_y$ for all $y \in \mathcal{X}$.

It is immediate to see that, conversely, if there exists W satisfying these properties (such a W is called a partial allocation state), then, from it, we can construct an allocation Q such that $W = W(Q)$. Clearly, under this correspondence, we have that $Q \in \mathcal{Q}$ iff W satisfies (P2) with equality for all $x \in \mathcal{X}$. In this case W is called an allocation state. The set of partial allocation states and the set of allocation states are denoted, respectively, with the symbols \mathcal{W}_p and \mathcal{W} . It is clear that two allocations Q^1 and Q^2 such that $W(Q^1) = W(Q^2)$, only differ for a permutation of the data atoms of the various units and for many purposes can be considered as equivalent. All the quantity of interest for the game theoretic setting will be defined at the level of W .

Allocation states are in general not unique:

Example 3. Suppose \mathcal{G} is the complete graph and that $\alpha_x = \alpha$, $\beta_x = \beta$ for every $x \in \mathcal{X}$, with $\alpha \leq \beta$. Among the possible allocation states there are those where each unit uses only one resource, namely for any permutation $\sigma : \mathcal{X} \rightarrow \mathcal{X}$ without fixed points, we can consider

$$W_{xy}^\sigma = \alpha \mathbb{1}_{\sigma(x)=y} \quad (4)$$

More generally, if s is an integer dividing α and we consider any symmetric s -regular graph over \mathcal{X} (without self-loops) with adjacency matrix A , we can consider the corresponding state allocation W^A given by

$$W_{xy}^A = \frac{\alpha}{s} A_{xy} \quad (5)$$

It is immediate to check that this is indeed a state allocation and that, in particular, $(W^A)^y = \alpha$ for every y .

In general, there are extra desired features that we may want solutions to possess:

- **(Reliability)** resources will often have different reliability properties and, in the allocation, a preference to more reliable resources should be given;
- **(Congestion)** equally reliable resources should be equally used, avoiding congestion phenomena;
- **(Aggregation)** units should use as few resources as possible to allocate their data.

Notice how in the example above, all the allocations states considered yield equal congestion for all resources while they differ from the point of view of aggregation: the solutions W^σ are of course the one maximizing this feature.

Instead of translating the above qualitative features into quantitative hard constraints on the allocation states, in this paper, we undertake a different road. We cast the problem into a game theoretic scenario defining, for each unit x and for each resource y , a function $f_{xy}(W)$ that is the utility of x in using y under the current (possibly partial) allocation state W . The utility functions are defined following the so called 'mechanism design' approach: utility will tend to be larger for more reliable or less congest resources or for resources where the unit has already allocated a larger size of its data. This will force the Nash equilibria of the game to be allocation states possessing, at a certain degree, the above features (typically in a compromised way depending on the parameter tuning of the utility functions). Our algorithm builds upon this game theoretic set-up. It is based on evolutionary game theory and is fully distributed: units iteratively allocate and move their data among the neighbors on the basis of the space available and trying to maximize a utility function that incorporates the features considered above. The algorithm always leads to an allocation, when such an allocation exists, that is a Nash equilibrium of the game.

2 The game theoretic set-up and the algorithm

In this section we present the details of the game theoretic structure and of the allocation algorithm.

Utilities are defined as follows. Under a (possibly partial) allocation state W , the utility of a unit x in using resource y is given by

$$f_{xy}(W) := \lambda_y - k_c W_y / \beta_y + k_a W_{xy} \quad (6)$$

The term λ_y encodes possible reliability differences among resources, the second term is a congestion term that takes into consideration the level of use of the resource, and, finally, the third term depends on both x and y and pushes a unit to allocate in those resources where it has already allocated. k_c, k_a are two non-negative parameters to tune the effect of the congestion and of the aggregation terms, respectively.

Given $W \in \mathcal{W}_p$ and $x \in \mathcal{X}$, put

$$\mathcal{X}^x(W) := \{y \in N_x \mid W_y < \beta_y\}$$

the set of resources still available for x under the current allocation state W . An allocation state $W \in \mathcal{W}$ (and also any $Q \in \mathcal{Q}$ such that $W(Q) = W$) is called a Nash equilibrium if, for every $x \in \mathcal{X}$, for every $y \in N_x$ such that $W_{xy} > 0$, for every $y' \in \mathcal{X}^x(W)$, it holds

$$f_{xy}(W) \geq f_{xy'}(W')$$

where $W' = W - e_{xy} + e_{xy'}$. Under a Nash equilibrium, a unit whose goal is to maximize its utility, has no advantage in moving their allocated data, under the standing assumption that only one data atom at a time can be moved. Notice that data atoms are to be interpreted as aggregations of data and the decision of their size is part of the design of the algorithm. Clearly different levels of granularity will give rise to different game models including different Nash equilibria.

Existence of Nash equilibria follow from the following considerations. Define

$$\Psi(W) := \sum_{y \in \mathcal{X}} \sum_{s=0}^{W_y} [\lambda_y - k_c s / \beta_y] + k_a \sum_{x, y \in \mathcal{X}} \sum_{s=0}^{W_{xy}} s \quad (7)$$

and notice that if $W, W' \in \mathcal{W}$ are such that $W' = W - e_{x\bar{y}} + e_{x\bar{y}'}$, then,

$$\begin{aligned} \Psi(W') - \Psi(W) &= (\lambda_{\bar{y}'} - k_c W'_{\bar{y}'} / \beta_{\bar{y}'} + k_a W'_{x\bar{y}'}) - (\lambda_{\bar{y}} - k_c W_{\bar{y}} / \beta_{\bar{y}} + k_a W_{x\bar{y}}) \\ &= f_{x\bar{y}'}(W') - f_{x\bar{y}}(W) \end{aligned} \quad (8)$$

In other terms, under the allocation state W , when user x moves a data atom from \bar{y} to \bar{y}' , he experiences a variation in utility given by $\Psi(W') - \Psi(W)$. Ψ is called a potential of the game. Maxima of Ψ are clearly Nash equilibria while, in general, the converse is not true. Considering that Ψ is defined on a finite set, a maximum, and thus a Nash equilibrium, always exists. The existence of a potential will be also fundamental in exploiting the properties of the allocation algorithm.

Remark (Relation with classical game theory): The game theoretic structure we have just defined is apparently different from the classical one: a set of players, each player playing an option and incurring in a utility that depends on the option he plays as well from the options played by the rest of the population. What are the players, the options and the utilities on our framework? One way to recast our model into a classical game theoretic one is that of considering as set of players not the set of units \mathcal{V} , but the set of data atoms to be stored \mathcal{A} . Given $(x, a) \in \mathcal{A}$, its set of options is simply given by N_x , the resources where it can be stored. A choice

of an option for each player is equivalent to an allocation Q . The utility of a player (x, a) under an allocation Q is given by $f_{xy}(W(Q))$. The game so defined is clearly potential with potential function given by Ψ .

A non-classical feature of our game is the presence of hard constraints ($W_y \leq \beta_y$) in the way options can be simultaneously played. The presence of such constraints have important consequences in the structure of Nash equilibria.

2.1 Examples

We now present a number of simple examples where Nash equilibria can be explicitly exhibited. The simplest case is when $k_a = 0$ since in this case the game becomes a pure congestion game.

Example 4. Suppose \mathcal{G} is the complete graph and that $\alpha_x = \alpha$, $\beta_x = \beta$, $\lambda_x = \lambda$ for every $x \in \mathcal{X}$ with $\alpha \leq \beta$. Suppose moreover that $k_a = 0$. It is immediate to check that any W such that $W^y = \alpha$ for every resource $y \in \mathcal{X}$ is a Nash equilibrium. We now check that these are all the possible ones. Indeed if W is a Nash equilibrium such that $W^y > \alpha > W^{y'}$ for some $y, y' \in \mathcal{X}$, it follows that

$$f_{xy}(W) = \frac{W^y}{\beta} < \frac{W^{y'} + 1}{\beta} = f_{xy'}(W + e_{xy'} - e_{xy})$$

This yields $W_{xy} = 0$ for every $x \neq y'$. Therefore, $W_{y'y} = W^y > \alpha$ that is a contradiction. Among the Nash equilibria considered above, there are those introduced in Example 3, W^σ and W^A . Since the potential Ψ depends on W only through the values $\{W^y\}$, it is easy to check that these Nash points are all maxima for Ψ .

In the case when \mathcal{G} is not complete, the existence of state allocations W where all resources are equally used (e.g. $W^y = \alpha$ for all y) is not longer guaranteed. From Theorem 1, a necessary and sufficient condition for this to happen is clearly $|D| \leq |N(D)|$ for every $D \subseteq \mathcal{X}$. Notice that when this condition is verified and we consider the bipartite graph $\tilde{\mathcal{G}} = (\mathcal{X} \times \mathcal{X}, \tilde{\mathcal{E}})$ where $(x, y) \in \tilde{\mathcal{E}}$ iff $(x, y) \in E$, Hall's theorem guarantees the existence of a matching in $\tilde{\mathcal{G}}$ complete on the first set, namely a permutation $\sigma : \mathcal{X} \rightarrow \mathcal{X}$ such that $(x, \sigma(x)) \in \tilde{\mathcal{E}}$ for every $x \in \mathcal{X}$. Using σ , as in the complete case, we can construct the state allocation W^σ where each unit uses just one resource. Note that the condition $|D| \leq |N(D)|$ is automatically verified in symmetric d -regular graphs. Indeed, if \mathcal{E}_D is the set of edges having one of the nodes in D , we have that $d|D| = |\mathcal{E}_D| \geq d|N(D)|$.

We now present an example where the condition $|D| \leq |N(D)|$ is violated.

Example 5. Let $\mathcal{X} = \{0, 1, 2\}^2$ and \mathcal{G} to be the 2-grid over \mathcal{X} , Suppose $\alpha_x = \alpha$, $\beta_x = \beta$, $\lambda_x = \lambda$ for every $x \in \mathcal{X}$ with $\alpha \leq \beta$. Suppose moreover that $k_a = 0$. Notice that $|D| \leq |N(D)|$ is not always verified. Indeed, if $D = \{(i, j) \mid 2 \mid (i + j)\}$ we have that $|D| = 5$ while $|N(D)| = 4$. This implies that allocations where all the resources are equally used can not exist. More precisely, it can not exist an allocation if $\beta < 5\alpha/4$. For $\beta \geq 5\alpha/4$ an allocation exists:

$$W_{(ij)(h,k)} = \begin{cases} \frac{\alpha}{2} & \text{if } i, j \in \{0, 2\} \\ \frac{\alpha}{4} & \text{if } (i, j) = (1, 1) \\ \frac{2\alpha}{5} & \text{if } 2 \nmid i + j, h, k \in \{0, 2\} \\ \frac{\alpha}{5} & \text{if } 2 \nmid i + j, (h, k) = (1, 1) \end{cases}$$

This yields: $W^{(i,j)} = 4\alpha/5$ if $(i,j) \in D$ while $W^{(i,j)} = 5\alpha/4$ if $(i,j) \in \mathcal{X} \setminus D$. It is immediate to verify that the neighbors of each unit are resources equally congested: this implies that W is a Nash equilibrium and, actually, for previous considerations, it is easy to show that is indeed a maximum of Ψ .

The case when $k_a > 0$ is generally harder to analyze. We start with a simple remark. Suppose $y_1, y_2 \in \mathcal{X}$ are two resources with the same affidability and storage capability β . Let W be a Nash equilibrium. If a unit x is using both resources under W (e.g. $W_{xy_i} > 0$ for $i=1,2$) and none of them is fully congested (e.g. $W^{y_i} < \beta$ for $i = 1, 2$), then the fact that W is Nash implies the double inequality:

$$\frac{W^{y_1} - W^{y_2}}{\beta} + k_a - \frac{1}{\beta} \leq k_a(W_{xy_1} - W_{xy_2}) \leq \frac{W^{y_1} - W^{y_2}}{\beta} - k_a + \frac{1}{\beta} \quad (9)$$

This implies that $k_a \leq 1/\beta$. Therefore, when $k_a > 1/\beta$, in a Nash equilibrium, a unit will never use more than one resource, unless all but one of them are fully congested.

Example 6. Suppose \mathcal{G} is the complete graph and that $\alpha_x = \alpha$, $\beta_x = \beta$, $\lambda_x = \lambda$ for every $x \in \mathcal{X}$ with $\alpha \leq \beta$. It is easy to show that the state allocations W defined in (4) are maxima of the potential also for any $k_a > 0$. In the case when $k_a \leq 1/\beta$, it follows from (9) that also the state allocations W^A proposed in (5) are still Nash equilibria (no longer maxima though if $k_a > 0$ and $s > 1$). Instead, in the case $k_a > 1/\beta$, it follows again from (9) that W^A is Nash only in the case $\beta = \alpha$, namely when resources are all fully congested. In the case $k_a > 1/\beta$, moreover, new Nash equilibria show up where resources have different congestion level. As a specific example, for the case $|\mathcal{X}| = 3$, the state allocation

$$W = \begin{pmatrix} 0 & 0 & \alpha \\ 0 & 0 & \alpha \\ \alpha & 0 & 0 \end{pmatrix}$$

is a Nash equilibrium for $\beta \geq 2\alpha$ and $k_a \geq 1/\beta$.

Even when the graph is complete and $k_a = 0$, the presence of resources with different affidability and the presence of hard storage constraints, may cause the presence of Nash equilibria where resources are not uniformly used. This is shown in the following example.

Example 7. Suppose $\mathcal{X} = \{1, 2, 3\}$ and \mathcal{G} is the complete graph (namely a 3-cycle). Values of α , β are given by

$$\begin{aligned} \alpha_1 &= 7 & \beta_1 &= 35 \\ \alpha_2 &= 35 & \beta_2 &= 21 \\ \alpha_3 &= 14 & \beta_3 &= 28 \end{aligned}$$

Suppose moreover that

$$\lambda_1 - 1 > \lambda_2 = \lambda_3 = 0$$

so that unit 1, as a resource, is always the best to use, even when it is fully congested. Of course, in any Nash equilibrium W , resource 1 will be saturated, therefore W will have the following structure

$$W = \begin{pmatrix} 0 & a & 7-a \\ b & 0 & 35-b \\ 35-b & b-21 & 0 \end{pmatrix}$$

with $a \in [0, 7] \cap \mathbb{N}$ and $b \in [21, 35] \cap \mathbb{N}$. Now notice that under a state like W above, units 2 and 3 are automatically at equilibrium, they have no convenience to move atoms. Let us analyze the utilities of unit 1. Defined $W' = W + e_{13} - e_{12}$ and $W'' = W + e_{12} - e_{13}$, we have that

$$\begin{aligned} f_{12}(W) &= \frac{21-a-b}{21}, & f_{13}(W') &= \frac{a+b-43}{28} \\ f_{13}(W) &= \frac{a+b-42}{28}, & f_{12}(W'') &= \frac{20-a-b}{21} \end{aligned}$$

Algebraic computations show that

$$\begin{aligned} f_{12}(W) &\geq f_{13}(W') &\Leftrightarrow a + b &\leq 30 \\ f_{13}(W) &\geq f_{12}(W'') &\Leftrightarrow a + b &\geq 30 \end{aligned}$$

We can thus conclude as follows: W is a Nash equilibrium if and only if the parameters a and b are in the following three ranges:

$$\begin{aligned} (I) \quad & 21 \leq b < 23 & a &= 7 \\ (II) \quad & 23 \leq b \leq 30 & a &= 30 - b \\ (III) \quad & 30 < b \leq 35 & a &= 0 \end{aligned}$$

Notice that in case (II), we have that $a + b = 30$: in this case resources 2 and 3 are equivalent in the sense that there is no convenience in moving data from one to the other and viceversa. Instead, in case (I) and (II), resources 2 and 3 give unbalanced utilities. Straightforward computation shows that, Nash equilibria of type II correspond to global maxima of the potential function ψ .

2.2 The algorithm

The allocation algorithm we are proposing is fully distributed and asynchronous and is only based on communications between units taking place along the links of the graph $\mathcal{G} = (\mathcal{X}, \mathcal{E})$.

At every instant of time, a unit is either in functional state on or off: we interpret off units, as units temporarily disconnected from the network, not available for communication and any action including storage and data retrieval. A unit, which is currently in state on, can activate and either allocate or move an atomic piece or its data among the available resources (e.g. those neighbors that still have place available and are on at that time). Choice of the available resource where to allocate will be following a classical stochastic relaxation of the utility maximization based on a Gibbs probability distribution: this is a classical choice in evolutionary game theory and will be akin to a fairly complete theoretical analysis.

We now describe all the mathematical details of the algorithm. We recall that a partial allocation is a function $Q : D \subseteq \mathcal{A} \rightarrow \mathcal{X}$ and that $W = W(Q)$ denotes the corresponding partial allocation state. The functional state of the network at a certain time will instead be denoted by $\xi \in \{0, 1\}^{\mathcal{X}}$: $\xi_x = 1$ means that the unit x is on. As already said, the basic ingredient of the algorithm is the Gibbs distribution that we now formally introduce. Given a functional state $\xi \in \{0, 1\}^{\mathcal{X}}$, a partial allocation state $W \in \mathcal{W}_p$ and a unit $x \in \mathcal{X}$, we define

$$\mathcal{X}^x(\xi, W) := \{y \in N_x \mid W_y < \beta_y, \xi_y = 1\}$$

that is the set of neighbors of x that are on and have still space available. We then denote with $p^x(\xi, W)$ the Gibbs probability distribution over $\mathcal{X}^x(\xi, W)$ given by

$$p_y^x(\xi, W) = \frac{e^{\gamma f_{xy}(W + e_{xy})}}{Z_\gamma^x(\xi, W)} \quad y \in \mathcal{X}^x(\xi, W)$$

where $\gamma > 0$ is a parameter and where

$$Z_\gamma^x(\xi, W) = \sum_{y \in \mathcal{X}^x(\xi, W)} e^{\gamma f_{xy}(W + e_{xy})}$$

is the normalizing factor.

The times when units modify their functional state (off to on or on to off) and the times when they activate to allocate/distribute are modeled as a family of independent Poisson clocks whose rates will be denoted (for unit x), respectively, ν_x^{on} , ν_x^{off} , and ν_x^{act} . The functional state of the network as a function of time $\xi(t)$ is thus a continuous time Markov process whose components are independent Bernoulli processes.

Mathematically, the algorithm can be described as a continuous-time process $Q(t) : D(t) \rightarrow \mathcal{X}$ taking values on the set of partial allocations \mathcal{Q}_p . We let $W(t) = W(Q(t))$ denote the corresponding process of partial allocation states.

The details of the algorithm are described below.

1. Assume unit \bar{x} activates at time t . With probabilities

$$P_{\bar{x}}^{\text{all}}(W(t)), \quad P_{\bar{x}}^{\text{dis}}(W(t)) = 1 - P_{\bar{x}}^{\text{all}}(W(t))$$

the resource \bar{x} will make, respectively, an allocation or a distribution move as explained below. Of course we are assuming that

$$\begin{aligned} P_{\bar{x}}^{\text{all}}(W(t)) &= 0 \text{ if } W(t)^{\bar{x}} = \alpha_{\bar{x}} \\ P_{\bar{x}}^{\text{dis}}(W(t)) &= 0 \text{ if } W(t)^{\bar{x}} = 0 \end{aligned}$$

2. (ALLOCATION MOVE)

- Choose (\bar{x}, \bar{a}) uniformly at random in $(D(t))^c \cap \mathcal{A}_{\bar{x}}$
- Choose \bar{y}' according to the Gibbs probability $p_{\bar{y}'}^{\bar{x}}(\xi(t), W(t))$
- Put $D(t^+) = D(t) \cup \{(\bar{x}, \bar{a})\}$,

$$Q(t^+)(x, a) = \begin{cases} \bar{y}' & \text{if } (x, a) = (\bar{x}, \bar{a}) \\ Q(t)(x, a) & \text{if } (x, a) \in D(t) \end{cases}$$

3. (DISTRIBUTION MOVE)

- Choose (\bar{x}, \bar{a}) uniformly at random in $D(t) \cap \mathcal{A}_{\bar{x}}$ and put $\bar{y} = Q(t)(\bar{x}, \bar{a})$
- Choose \bar{y}' according to the Gibbs probability $p_{\bar{y}'}^{\bar{x}}(\xi(t), W(t) - e_{\bar{x}\bar{y}})$
- Put $D(t^+) = D(t)$,

$$Q(t^+)(x, a) = \begin{cases} \bar{y}' & \text{if } (x, a) = (\bar{x}, \bar{a}) \\ Q(t)(x, a) & \text{if } (x, a) \in D(t) \setminus \{(\bar{x}, \bar{a})\} \end{cases}$$

3 Analysis of the algorithm

In this section we analyze the behavior of the algorithm introduced above. We will essentially show two results. First, we prove that if the set of allocation \mathcal{Q} is not empty (i.e. condition (1) is satisfied), the algorithm above will reach such an allocation in bounded time with probability 1 (e.g $D(t) = \mathcal{A}$ for t sufficiently large).

Second, we will show that, under a slightly stronger assumption than (1), in the double limit $t \rightarrow +\infty$ and then $\gamma \rightarrow +\infty$, the process $Q(t)$ induced by the algorithm will always converge, in law, to a Nash equilibrium that is a global maximum of the potential function Ψ .

In order to prove such results, it will be necessary to go through a number of intermediate technical steps. First of all, it will be convenient to work directly with the process $W(t) = W(Q(t))$. By the fact that the jump probabilities of the original process $Q(t)$ are only function of $W(t)$ and of $\xi(t)$, it immediately follows that the augmented process $(\xi(t), W(t))$ is Markovian and its only non zero transition rates are described below:

$$\Lambda_{(\xi, W), (\xi', W)} = \begin{cases} \nu_{\bar{x}}^{on} & \text{if } \xi_{\bar{x}} = 0, \xi'_{\bar{x}} = 1, \xi_x = \xi'_x \forall x \neq \bar{x} \\ \nu_{\bar{x}}^{off} & \text{if } \xi_{\bar{x}} = 1, \xi'_{\bar{x}} = 0, \xi_x = \xi'_x \forall x \neq \bar{x} \end{cases} \quad (10)$$

$$\Lambda_{(\xi, W), (\xi, W')} = \begin{cases} \nu_{\bar{x}}^{act} P_{\bar{x}}^{\text{all}}(W) p_{\bar{y}'}^{\bar{x}}(\xi, W) & \text{if } \xi_{\bar{x}} = \xi_{\bar{y}'} = 1, W' = W + e_{\bar{x}\bar{y}'} \\ \nu_{\bar{x}}^{act} P_{\bar{x}}^{\text{dis}}(W) \frac{W_{\bar{x}\bar{y}}}{\alpha_{\bar{x}}} p_{\bar{y}'}^{\bar{x}}(\xi, W - e_{\bar{x}\bar{y}}) & \text{if } \xi_{\bar{x}} = \xi_{\bar{y}} = \xi_{\bar{y}'} = 1, W' = W + e_{\bar{x}\bar{y}'} - e_{\bar{x}\bar{y}} \end{cases} \quad (11)$$

We now introduce a graph on \mathcal{W}_p that will be denoted by \mathcal{L}_p : an edge (W, W') is present in \mathcal{L}_p if and only if $\Lambda_{(\mathbf{1}, W), (\mathbf{1}, W')} > 0$.

Our strategy will be to show that from any element $W \in \mathcal{W}_p$ there is a path in \mathcal{L}_p to some element $W' \in \mathcal{W}$. Considering that there is a non-zero probability that the process $(\xi(t), W(t))$ starting from (ξ, W) will reach $(\mathbf{1}, W)$ in bounded time (because of the presence of transitions (10)), standard probabilistic arguments will then allow to conclude that allocation (e.g. $D(t) = \mathcal{A}$) will be achieved in bounded time with probability 1.

Given $W \in \mathcal{W}_p$ we define the following subsets of units

$$\mathcal{X}^f(W) := \{x \in \mathcal{X} \mid W^x = \alpha_x\},$$

$$\mathcal{X}^{sat}(W) := \{x \in \mathcal{X} \setminus \mathcal{X}^f(W) \mid \nexists y \in N_x \text{ s.t. } W(Q)_y < \beta_y\}$$

Units in $\mathcal{X}^f(W)$ are called fully allocated: these units have completed the allocation of their data under the state W . Units in $\mathcal{X}^{sat}(W)$ are called saturated: they have not yet completed their allocation, however, under the current state W , they can not make any action, neither allocate, nor distribute. Finally, define

$$\mathcal{W}_p^{sat} := \{W \in \mathcal{W}_p \setminus \mathcal{W} \mid \mathcal{X} = \mathcal{X}^f(W) \cup \mathcal{X}^{sat}(W)\}$$

It is clear that from any $W \in \mathcal{W}_p \setminus \mathcal{W}_p^{sat}$, some allocation move can be performed. Instead, if we are in a state $W \in \mathcal{W}_p^{sat}$, only possibly fully allocated units can make a distribution move. Notice that, because of condition (1), for sure there exist resources y such that $W_y < \beta_y$ and these resources are indeed exclusively connected to fully allocated units. The key point is to show that in a finite number of distribution moves it is always possible to move some data atoms from resources connected to saturated units to resources with available space: this will then make possible a further allocation move.

For any fixed $W \in \mathcal{W}_p$, we can consider the following graph structure on \mathcal{X} thought as set of resources: $\mathcal{H}_W = (\mathcal{X}, \mathcal{E}_W)$. Given $y_1, y_2 \in \mathcal{X}$, there is an edge from y_1 to y_2 if and only if there exists $x \in \mathcal{X}$ for which

$$W_{xy_1} > 0, \quad (x, y_2) \in \mathcal{E}$$

The edge from y_1 to y_2 will be indicated with the symbol $y_1 \rightarrow_x y_2$ (to also recall the unit x involved). The presence of the edge means that the two resources y_1 and y_2 are in the neighborhood of a common unit x that is using y_1 under W . This indicates that x can in principle move some of its data currently stored in y_1 into resource y_2 if this last one is available. We have the following technical result

Lemma 2. *Suppose $(\mathcal{G}, \alpha, \beta)$ satisfies (1). Fix $W \in W_p$ and let $\bar{y} \in \mathcal{X}$ be such that there exists $\bar{x} \in N_{\bar{y}}$ with $W^{\bar{x}} < \alpha_{\bar{x}}$. Then, there exists a sequence*

$$\bar{y} = y_0, x_0, y_1, \dots, y_{t-1}, x_{t-1}, y_t \quad (12)$$

satisfying the following conditions

- (Sa) *Both families of the y_k 's and of the x_k 's are each made of distinct elements;*
- (Sb) *$y_k \rightarrow_{x_k} y_{k+1}$ for every $k = 0, \dots, t-1$;*
- (Sc) *$W_{y_k} = \beta_{y_k}$ for every $k = 0, \dots, t-1$, and $W(Q)_{y_t} < \beta_{y_t}$.*

Proof Let $\mathcal{Y} \subseteq \mathcal{X}$ be the subset of nodes that can be reached from \bar{y} in \mathcal{H}_W . Preliminarily, we prove that there exists $y' \in \mathcal{Y}$ such that $W_{y'} < \beta_{y'}$. Let

$$\mathcal{Z} := \{x \in \mathcal{X} \mid \exists y \in \mathcal{Y}, W_{xy} > 0\}$$

and notice that, by the way \mathcal{Y} and \mathcal{Z} have been defined,

$$x \in \mathcal{Z}, (x, y) \in \mathcal{E} \Rightarrow y \in \mathcal{Y} \quad (13)$$

Suppose now that, contrarily to the thesis, $W_y \geq \beta_y$ for all $y \in \mathcal{Y}$. Then,

$$\sum_{x \in \mathcal{Z}} \alpha_x \leq \sum_{y \in \mathcal{Y}} \beta_y = \sum_{y \in \mathcal{Y}} W_y = \sum_{y \in \mathcal{Y}} \sum_{x \in \mathcal{Z}} W_{xy} = \sum_{x \in \mathcal{Z}} W^x < \sum_{x \in \mathcal{Z}} \alpha_x \quad (14)$$

where the first inequality follows from (13) and (1), the first equality from the contradiction hypothesis, the second equality from the definition of \mathcal{Z} , the third equality again from (13) and, finally, last inequality from the existence of \bar{x} . This is clearly absurd and thus proves our claim.

Consider now a path of minimal length from \bar{y} to \mathcal{Y} in \mathcal{H}_W :

$$\bar{y} = y_0 \rightarrow_{x_0} y_1 \rightarrow_{x_1} \dots \rightarrow_{x_{t-2}} y_{t-1} \rightarrow_{x_{t-1}} y_t$$

and notice that the sequence $\bar{y} = y_0, x_0, y_1, \dots, y_{t-1}, x_{t-1}, y_t$ will automatically satisfy properties (Sa) to (Sc). ■

We are now ready to prove the first main result.

Theorem 3. *Assume that*

1. $\nu_x^{on} > 0$ and $\nu_x^{act} > 0$ for every $x \in \mathcal{X}$,
2. $P_x^{all}(W, \bar{x}) > 0$ if $W^x < \alpha_x$,
3. $(\mathcal{G}, \alpha, \beta)$ satisfies (1).

Then,

$$\mathbb{P}(\exists t_0 \mid W(t) \in \mathcal{W} \forall t \geq t_0) = 1$$

Proof In order to prove the claim, it will be sufficient to show that from any $W \in \mathcal{W}_p$ there is a path in \mathcal{L}_p to some element $W' \in \mathcal{W}$. We will prove it by a double induction process. To this aim we consider two indices associated to any $W \in \mathcal{W}_p \setminus \mathcal{W}$. The first one is defined by

$$m_W = \sum_{x \in \mathcal{X}} (\alpha_x - W^x) \geq 1$$

To define the second, consider any $\bar{x} \in \mathcal{X} \setminus \mathcal{X}^f(W)$. We can apply Lemma 2 to W and any $\bar{y} \in N_{\bar{x}}$ and obtain that we can find a sequence of agents $\bar{y} = y_0, x_0, y_1, \dots, y_{t-1}, x_{t-1}, y_t$ satisfying the properties (Sa), (Sb), and (Sc) above. Among all the possible choices of $\bar{x} \in \mathcal{X}$, $\bar{y} \in N_{\bar{x}}$ and of the corresponding sequence, assume we have chosen the one minimizing t and denote such minimal t by t_W . The induction process will be performed with respect to the lexicographic order induced by the pair (m_W, t_W) .

In the case when $t_W = 0$, it means we can find $\bar{x} \in \mathcal{X}$, $\bar{y} \in N_{\bar{x}}$ such that $W_{\bar{y}} < \beta_{\bar{y}}$. Therefore, under the allocation state W , the unit \bar{x} can allocate a further data atom to \bar{y} . Considering that the activation of the unit \bar{x} has positive probability because of assumptions 1. and 2., this shows that W is connected to a W' such that $m_{W'} < m_W$. In case $m_W = 1$, this means that $W' \in \mathcal{W}$.

Consider now any $W \in \mathcal{W}_p \setminus \mathcal{W}$ such that $t = t_W > 1$. Let $\bar{x} \in \mathcal{X}$, $\bar{y} \in N_{\bar{x}}$ and the sequence $\bar{y} = y_0, x_0, y_1, \dots, y_{t-1}, x_{t-1}, y_t$ satisfying the properties (Sa), (Sb), and (Sc) above. In the allocation state W , the unit x_{t-1} , if activated (and this again has positive probability to happen because of assumptions 1. and 2.), can thus move an atomic piece of data from y_{t-1} to y_t . The new allocation state is $W' = W - e_{x_{t-1}y_{t-1}} + e_{x_{t-1}y_t}$. Since $W'_{y_{t-1}} < \beta_{y_{t-1}}$, for sure $t_{W'} < t_W$. The induction argument is thus complete. ■

We are now left with studying the Markov process $W(t)$ on \mathcal{W} . Noisy Gibbs best response dynamics are known to yields reversible Markov processes. This is indeed the case also in our case once the process has reached the set of allocations \mathcal{W} . Precisely, the following result holds:

Proposition 4. $(\xi(t), W(t))$, restricted to $\{0, 1\}^{\mathcal{X}} \times \mathcal{W}$, is a time reversible Markov process. More precisely, for every $(\xi, W), (\xi', W') \in \{0, 1\}^{\mathcal{X}} \times \mathcal{W}$ it holds

$$\rho_{(\xi, W)} \Lambda_{(\xi, W), (\xi', W')} = \rho_{(\xi', W')} \Lambda_{(\xi', W'), (\xi, W)} \quad (15)$$

where

$$\rho_{(\xi, W)} = \left[\prod_{x: \xi_x=1} \nu_x^{\text{on}} \prod_{x: \xi_x=0} \nu_x^{\text{off}} \right] \frac{\prod_x \alpha_x!}{\prod_{x,y} W_{xy}!} e^{\gamma \Psi(W)} \quad (16)$$

Proof It follows from relations (10) and (11) that the only cases when $\Lambda_{(\xi, W), (\xi', W')}$ and $\Lambda_{(\xi', W'), (\xi, W)}$ are not both equal to zero are the following:

- (i) $W' = W$, $\xi_{\bar{x}} = 0$, $\xi'_{\bar{x}} = 1$, $\xi_x = \xi'_x \forall x \neq \bar{x}$
- (ii) $W' = W$, $\xi_{\bar{x}} = 1$, $\xi'_{\bar{x}} = 0$, $\xi_x = \xi'_x \forall x \neq \bar{x}$
- (iii) $\xi' = \xi$, $W' = W + e_{\bar{x}\bar{y}'} - e_{\bar{x}\bar{y}}$.

In case (i), we have that

$$\frac{\rho(\xi, W)}{\rho(\xi', W')} = \frac{\prod_{x: \xi_x=1} \nu_x^{\text{on}} \prod_{x: \xi_x=0} \nu_x^{\text{off}}}{\prod_{x: \xi'_x=1} \nu_x^{\text{on}} \prod_{x: \xi'_x=0} \nu_x^{\text{off}}} = \frac{\nu_x^{\text{off}}}{\nu_x^{\text{on}}} = \frac{\Lambda(\xi', W'), (\xi, W)}{\Lambda(\xi, W), (\xi', W')}$$

Case (ii) can be analogously verified.

Consider now case (iii). We have that:

$$\frac{\rho(\xi, W)}{\rho(\xi', W')} = \frac{W'_{\bar{x}\bar{y}'}}{W_{\bar{x}\bar{y}}} e^{\gamma(\Psi(W) - \Psi(W'))}$$

On the other hand, using relation (11) and the fact that $Z_{\gamma}^{\bar{x}}(\xi, W) = Z_{\gamma}^{\bar{x}}(\xi', W')$, we obtain

$$\frac{\Lambda(\xi', W'), (\xi, W)}{\Lambda(\xi, W), (\xi', W')} = \frac{W'_{\bar{x}\bar{y}'}}{W_{\bar{x}\bar{y}}} e^{\gamma(f_{\bar{x}\bar{y}}(W) - f_{\bar{x}\bar{y}'}(W'))}$$

We can now conclude using relation (8). ■

We now show that under a slight stronger assumption than (1), namely,

$$\sum_{x \in A} \alpha_x < \sum_{y \in N(A)} \beta_y \quad \forall A \subseteq \mathcal{X}, \quad (17)$$

the process $(\xi(t), W(t))$ restricted to $\{0, 1\}^{\mathcal{X}} \times \mathcal{W}$ is ergodic. Denote by \mathcal{L} the graph \mathcal{L}_p restricted to the set \mathcal{W} . Notice that, as a consequence of time-reversibility, \mathcal{L} is an undirected graph. Ergodicity is equivalent to proving that \mathcal{L} is connected. We start with a lemma analogous to previous Lemma 2.

Lemma 5. *Suppose $(\mathcal{G}, \alpha, \beta)$ satisfies (17) and let $W \in \mathcal{W}$. Then, for every $\bar{y} \in \mathcal{X}$, there exists a sequence (12) satisfying the conditions (Sa), (Sb), and (Sc) as in Lemma 2.*

Proof It is sufficient to follow the steps of to the proof of Lemma 2 noticing that in (14) the first equality is now a strict inequality, while the last strict inequality becomes an equality. ■

If $W, W' \in \mathcal{W}$ are connected through a path in \mathcal{L} , we write that $W \sim W'$. Introduce the following distance on \mathcal{W} : if $W^1, W^2 \in \mathcal{W}$

$$\delta(W^1, W^2) = \sum_{x, y} |W_{xy}^1 - W_{xy}^2|$$

A pair $(W^1, W^2) \in \mathcal{W}$ is said to be minimal if

$$\delta(W^1, W^2) \leq \delta(W^{1'}, W^{2'}) \quad \forall W^{1'} \sim W^1, \forall W^{2'} \sim W^2$$

Notice that \mathcal{L} is connected if and only if for any minimal pair (W^1, W^2) , it holds $W^1 = W^2$.

Lemma 6. *Let (W^1, W^2) be a minimal pair. Suppose $y \in \mathcal{X}$ is such that $W_y^1 < \beta_y$. Then, $W_{xy}^1 = W_{xy}^2$ for all $x \in \mathcal{X}$.*

Proof Suppose by contradiction that $W_{xy}^1 < W_{xy}^2$ for some $x \in \mathcal{X}$. Then, necessarily, there exists $y' \neq y$ such that $W_{xy'}^1 > W_{xy'}^2$. Consider then $W^{1'} = W^1 - e^{xy'} + e^{xy}$. Clearly, $\delta(W^{1'}, W^2) < \delta(W^1, W^2)$ and this contradicts the minimality assumption. Thus $W_{xy}^1 \geq W_{xy}^2$ for all $x \in \mathcal{X}$. This yields $W_y^2 < \beta_y$. Exchanging the role of W^1 and W^2 we obtain the thesis. \blacksquare

Proposition 7. *If condition (17) holds true, the graph \mathcal{L} is connected.*

Proof Let (W^1, W^2) be any minimal pair. We will prove that W^1 and W^2 are necessarily identical. Consider any resource y . It follows from Lemma 5 that we can find a sequence $y = y_0, x_0, y_1, \dots, y_{t-1}, x_{t-1}, y_t$ satisfying the same (Sa), (Sb), and (Sc) with respect to the state allocation W^1 . Among all the possible sequences, choose one with t minimal for given y . We will prove by induction on t that $W_{xy}^1 = W_{xy}^2$ for all $x \in \mathcal{X}$.

If $t = 0$, it means that $W_y^1 < \beta_y$. It then follows from Lemma 6 that $W_{xy}^1 = W_{xy}^2$ for all $x \in \mathcal{X}$. Suppose now that the claim has been proven for all minimal pairs (W^1, W^2) and any $y \in \mathcal{X}$ for which $t < \bar{t}$ (w.r. to W^1) and assume that $y = y_0, x_0, y_1, \dots, y_{\bar{t}-1}, x_{\bar{t}-1}, y_{\bar{t}}$ satisfies the properties (Sa), (Sb), and (Sc) with respect to W^1 . Notice that the unit $x_{\bar{t}-1}$ can move a data atom from resource $y_{\bar{t}-1}$ into resource $y_{\bar{t}}$ under the state allocation W^1 and obtain $W^{1'} = W^1 - e^{x_{\bar{t}-1}y_{\bar{t}-1}} + e^{x_{\bar{t}-1}y_{\bar{t}}}$. Consider now W^2 and notice that Lemma 6 yields $W_{x_{\bar{t}-1}y_{\bar{t}}}^2 = W_{x_{\bar{t}-1}y_{\bar{t}}}^1 < \beta_{y_{\bar{t}}}$. Define

$$W^{2'} = \begin{cases} W^2 & \text{if } W_{x_{\bar{t}-1}y_{\bar{t}-1}}^2 = 0 \\ W^2 - e^{x_{\bar{t}-1}y_{\bar{t}-1}} + e^{x_{\bar{t}-1}y_{\bar{t}}} & \text{if } W_{x_{\bar{t}-1}y_{\bar{t}-1}}^2 > 0 \end{cases}$$

Clearly, $\delta(W^{1'}, W^{2'}) \leq \delta(W^1, W^2)$ and this implies that also $W^{1'}, W^{2'}$ is a minimal pair. Notice that $y = y_0, x_0, y_1, \dots, y_{\bar{t}-1}$ satisfies (Sa), (Sb), and (Sc) with respect to $W^{1'}$. Therefore, by the induction hypotheses, it follows that $W_{xy}^{1'} = W_{xy}^{2'}$ for all $x \in \mathcal{X}$. Since $W_{xy}^1 = W_{xy}^{1'}$ and $W_{xy}^2 = W_{xy}^{2'}$, result follows immediately. \blacksquare

Propositions 4 and 7 immediately yield the following final result.

Corollary 8. *Suppose that $(\mathcal{G}, \alpha, \beta)$ satisfies (17). Then $(\xi(t), W(t))$, restricted to $\{0, 1\}^{\mathcal{X}} \times \mathcal{W}$, is an ergodic time reversible Markov process whose unique invariant probability measure is given by*

$$\mu_\gamma(\xi, W) = Z_\gamma^{-1} \left[\prod_{x: \xi_x=1} \nu_x^{\text{on}} \prod_{x: \xi_x=0} \nu_x^{\text{off}} \right] \binom{\alpha}{W} e^{\gamma \Psi(W)}$$

where Z_γ is the normalizing constant.

Remark: It follows from previous result that the process $W(t)$ converges in law to the probability distribution

$$\tilde{\mu}_\gamma(W) := \tilde{Z}_\gamma^{-1} \binom{\alpha}{W} e^{\gamma \Psi(W)}$$

Notice that when $\gamma \rightarrow +\infty$, the probability $\tilde{\mu}_\gamma$ converges to a probability concentrated on the set $\text{argmax}_{W \in \mathcal{W}} \Psi(W)$ of state allocations maximizing the potential, thus on particular Nash equilibria. Thus, if γ is small, the distribution of the process $W(t)$ for t sufficiently large will be close to a Nash equilibrium.

Remark: Condition (17) is necessary for ergodicity. Notice indeed that in the case when \mathcal{G} is complete and $\alpha_x = \beta_x = a$ for all $x \in \mathcal{X}$, under every allocation W such that $W^y = a$ for every y , all resources will be saturated and, consequently, no distribution move will be allowed in W . Such allocations W are thus all sinks in the graph \mathcal{L} that is therefore not connected.

4 Simulation

In this section, we present a number of numerical simulations that validate our theoretical results and show the behavior of the proposed algorithm. We analyze the structure of the asymptotic Nash configuration reached and the effect of the variations of the parameters in the utility function. The algorithm is tested for its convergence properties and evaluated through some performance indices measuring the computational and topological complexity of the solution found.

Below we report the basic assumptions of our simulations.

- Most of our simulations will be for $n = 10$ or $n = 50$ units while larger populations will be considered to show the scalability of the algorithm.
- We consider two possible underlying networks, a complete graph and a regular graph with degree 10.
- We consider the case of homogeneous populations where all units have the same α_x , β_x and λ_x , as well heterogeneous cases with two subpopulations \mathcal{X}_1 and \mathcal{X}_2 sharing different parameters.
- In the utility function we set the congestion parameter to be $k_c = 1$ while we consider different values for the aggregation parameter k_a .
- Most of our simulations will assume that units are always on. An example where units may be in off mode is however presented at the end.
- In the implementation of the algorithm, time is supposed to be discrete, assuming that at each instant of time a unit x wakes up with a probability proportional to α_x . The allocation and distribution moves are chosen following the probabilities:

$$P_{\text{all}}(W(t), x) = \begin{cases} 1 & \text{if } W(t)^x < \alpha_x \\ 0 & \text{otherwise} \end{cases}, \quad P_{\text{dis}}(W(t), x) = 1 - P_{\text{all}}(W(t), x)$$

- The parameter γ in the Gibbs distribution is chosen to be time-varying with the law:

$$\gamma(t) = \gamma(t-1) + \frac{1}{\lambda_{\max} * 100}$$

where λ_{\max} is the maximum reliability of the resources.

- The typical time horizon we will consider in the algorithm will be $T = 10 \sum_x \alpha_x$: in this way every unit x will activate, on the average, a number of times equal to 10 times the number of data atoms it needs to allocate. As we will see, this time range is in general sufficient to reach, or at least get close, to a Nash equilibrium. Greater time ranges will be considered in some cases.

For all examples, the performance of the algorithm will be analyzed considering the following parameters computed, in a Montecarlo style, by averaging over 10 running of the algorithm.

- **Distance from optimum:** $\psi = \frac{\Psi(W^T)}{\Psi_{opt}}$ where Ψ_{opt} is the maximum of the potential. This allows to measure how the final allocation state $W(T)$ is far from a Nash configuration maximizing the potential Ψ . We will use the exact value of Ψ_{opt} in the homogeneous cases treated in Examples 4 and 6 for which the optimum Nash is explicitly known. For the heterogeneous cases we will instead approximate it using standard Matlab numerical algorithm.

In some cases we will also present a plot of the function $\psi(t) = \frac{\Psi(W(t))}{\Psi_{opt}}$ in this case for a single running of the algorithm.

- **Allocation complexity:** the parameter

$$\nu_{moves} = \frac{1}{n} \sum_{x \in \mathcal{X}} \frac{m_x}{\alpha_x}$$

where m_x is the total number of moves (allocation and distribution) made by unit x . ν_{moves} measures the number of allocation and distribution moves per piece of data. Since moving data from a resource to another can be expensive, it is an important parameter to control.

- **Average satisfaction:**

$$\bar{\Lambda} := \frac{1}{n} \sum_{x \in \mathcal{X}} \sum_{y \in N_x} \frac{W_{xy}^T}{\alpha_x} \lambda_y$$

This is meaningful when there are resources with different values for λ_y . If we interpret λ_y as a measure of the quality of a resource y (reflecting possible different features as responsiveness, security, etc), $\bar{\Lambda}$ measures the efficiency of the algorithm to choose the best resources where to allocate.

- **Topological complexity:** We consider the subgraph constituted by the edges (x, y) such that $W_{xy}^T > 0$ and we compute the average out-degree and average in-degree for the two subpopulations in the heterogeneous case

$$d^+ := \frac{1}{n} \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{X}} \mathbb{1}_{\{W_{xy}^T > 0\}}, \quad d_i^- := \frac{2}{n} \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{X}_i} \mathbb{1}_{\{W_{xy}^T > 0\}}, \quad i = 1, 2$$

d^+ measures the way units have split their data among the available neighbors, while d_i^- the average amount of units using a resource of a certain type.

- **Average congestion:** For the heterogeneous case, the following index measure the congestion of the two classes of resources:

$$\bar{C}^i := \frac{1}{n\beta} \sum_{y \in \mathcal{X}_i} W_y^T$$

4.1 Homogeneous case

In this section we consider the case when all units have equal parameters: $\alpha_x = \alpha$, $\beta_x = \beta$ and $\lambda_x = \lambda$ for every $x \in \mathcal{X}$. In this case Nash equilibria were studied in Examples 4 and 6. In particular, we know that, for any value of the aggregation parameter k_a , the maximum of the potential is given by the Nash equilibria where each unit uses just one resource: $W_{xy}^\sigma = \alpha \mathbb{1}_{\sigma(x)=y}$ where $\sigma : \mathcal{X} \rightarrow \mathcal{X}$ is any permutation without fixed points.

Example 8. Assume that $n = 10$ and that the underlying graph is complete. Parameters are set to the values: $\alpha = 27$, $\beta = 30$, and $\lambda = 3$. We consider the following values for the aggregation term k_a : 0, 0.003, 0.1. Notice that only case when $k_a > 1/\beta$ is for $k_a = 0.1$: in this case we expect to find final state allocation where the aggregation feature prevails according to the considerations in Example 6.

For all simulations we compute the indices ψ , ν_{moves} and d^+ . Here are the values we have obtained:

	$k_a = 0$	$k_a = 0.003$	$k_a = 0.1$
ν_{moves}	8.7863	6.2907	3.5359
d^+	7.4100	1	1
ψ	0.9994	1	1

Table 1: Values for $k_a = 0$, $k_a = 0.003$, $k_a = 0.1$ in the complete case

Notice how the number of moves per atom is decreasing with k_a : this feature is due to the fact that the stronger is the aggregation term, the more the initial allocation will tend to immediately create a solution with aggregation features already close to a Nash equilibrium.

Notice how for $k_a = 0$ the optimal is in general not reached, while it is reached in the case $k_a = 0.003$ and $k_a = 0.1$. In order to better understand the convergence behavior, below we present, for a single running of the algorithm, the plot of $\psi(t)$:

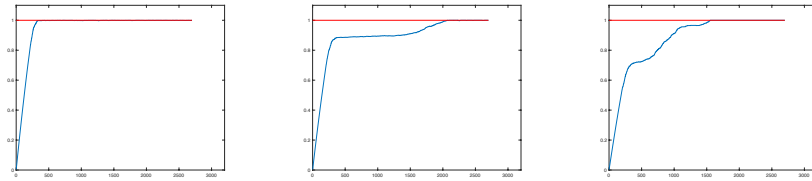


Figure 1: Plot of $\psi(t)$ for $k_a = 0$, $k_a = 0.003$, $k_a = 0.1$

Notice how, in the case $k_a = 0$, the convergence is indeed fast in the initial part but then it slows down and the dynamics remains for a long time fluctuating in matrices close to a Nash. A typical running of the algorithm yields a final configuration

$W(T)$ of type

$$W(T) = \begin{pmatrix} 0 & 4 & 2 & 2 & 1 & 1 & 5 & 8 & 2 & 2 \\ 4 & 0 & 8 & 3 & 3 & 2 & 4 & 1 & 1 & 1 \\ 3 & 2 & 0 & 2 & 4 & 1 & 1 & 5 & 4 & 5 \\ 1 & 5 & 5 & 0 & 1 & 0 & 0 & 10 & 3 & 2 \\ 6 & 0 & 3 & 1 & 0 & 5 & 3 & 2 & 4 & 3 \\ 2 & 1 & 2 & 1 & 2 & 0 & 6 & 0 & 7 & 6 \\ 3 & 4 & 0 & 5 & 7 & 3 & 0 & 1 & 1 & 3 \\ 2 & 5 & 2 & 4 & 4 & 1 & 3 & 0 & 2 & 4 \\ 3 & 5 & 4 & 2 & 2 & 4 & 1 & 3 & 0 & 3 \\ 5 & 0 & 3 & 6 & 3 & 5 & 3 & 0 & 2 & 0 \end{pmatrix}$$

with congestion vector $C = \mathbf{1} * W(T) = (29 \ 26 \ 29 \ 26 \ 27 \ 22 \ 26 \ 30 \ 26 \ 29)$. In this case, in order to achieve the optimum with all resources equally congested, the algorithm must run for much longer time, order of $200 * \sum_x \alpha_x$. This is a peculiarity of the case without aggregation term and such phenomena never shows up otherwise. A faster convergence however can be obtained in this case by a different tuning of the noise function. Using

$$\gamma(t) = \gamma(t-1) + \frac{1}{\lambda_{max} * 10}$$

we typically reach an optimum at $T = 30 * \sum_x \alpha_x$.

In the case when $k_a = 0.003$ or $k_a = 0.1$ convergence is slower in the initial part, however, the typical state allocation $W(T)$ constructed by the algorithm will instead be a matrix of type W^σ , namely an optimal Nash equilibrium.

The aim of the following example is to prove that the different non-optimal Nash equilibria presented in Example 6 can attract the dynamics of the algorithm.

Example 9. Assume that $n = 10$ and that the underlying graph is complete. Parameters are set to the values: $\alpha = 27$, $\beta = 60$ (so that $\beta > 2\alpha$), $\lambda = 3$, and $k_a = 0.1 > 1/\beta$. The time horizon is fixed to $T = 20 * \sum_x \alpha_x$. Here are the values of ψ , ν_{moves} and d^+ obtained in this case:

ν_{moves}	3.2781
$d^+ = d^-$	1
ψ	0.9805

Table 2: Values for $\beta > 2\alpha$

Notice how the optimal Nash (e.g. any aggregated matrix W^σ) is not reached in general. If we analyze single runnings of the algorithm, we can verify that the algorithm reach different possible Nash equilibria: besides the optimal W^σ , we observe also final configurations where one or more resources are used by two units. Here is a sample of final configurations observed

$$W = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 27 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 27 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 27 \\ 27 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 27 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 27 & 0 \\ 0 & 0 & 0 & 0 & 27 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 27 \\ 0 & 0 & 27 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 27 & 0 & 0 & 0 \end{pmatrix}$$

Notice how such Nash equilibria were indeed correctly forecasted in Example 6.

We conclude the homogeneous case with an example showing the good scalability properties of the algorithm.

Example 10. We consider the case when $n = 50, 100, 1000$ and that the underlying graph is a random regular graph with degree 10 (the use of the complete graph for large number of units becomes heavy on the computational side and, in any case, is also of little practical interest). Parameters are set to the values: $\alpha = 27$, $\beta = 30$, $\lambda = 3$, $k_a = 0.03$. Here are the values obtained for ψ , ν_{moves} and d^+ :

	$n = 50$	$n = 100$	$n = 1000$
ν_{moves}	2.6876	2.7064	1.4592
$d^+ = d^-$	1	1.0870	1.1000
ψ	1	0.9978	0.9973

Table 3: Scalability of the Algorithm

4.2 Non-homogeneous case

In this section we consider the case when all units have the same $\alpha_x = \alpha$ and $\beta_x = \beta$ but, for what the reliability is concerned, the population is split into two parts \mathcal{X}_1 and \mathcal{X}_2 and $\lambda_x = \lambda^i$ if $x \in \mathcal{X}_i$ with $\lambda^1 \neq \lambda^2$. While we do not have in this case any explicit computation of the optimal Nash equilibrium, it can easily be shown that the aggregate solution W^σ is a Nash Equilibrium (possibly not a maximum of the potential) if and only if $k_a \geq \frac{\lambda^2 - \lambda^1 - 1/\beta}{\alpha - 1}$.

Example 11. Assume that $n = 50$, $|\mathcal{X}_1| = |\mathcal{X}_2| = 25$, and that the underlying graph is either complete or random regular of degree 10. Parameters are set to the values: $\alpha = 45$, $\beta = 50$, $\lambda^1 = 0.5$, $\lambda^2 = 0.8$. We consider different values for the aggregation term k_a . For all simulations we compute the parameters ψ , ν_{moves} , d^+ , d^- , $\bar{\Lambda}$, and \bar{C}^i averaged over 10 running of the algorithm. Here are the values we have obtained in the two cases of a complete graph and of a random regular graph of degree 10.

These results, on one hand, show how the speed and computational performance of the algorithm in these heterogeneous cases are similar to the homogeneous case: at the horizon time T the system is always very close to (or is exactly at) the optimal solution and the number of moves per atom is similar. On the other hand, other interesting features emerge from an analysis of the other parameters. The congestion parameters show how, for very small aggregation term $k_a = 0.001$, the most reliable

	$k_a = 0.001$	$k_a = 0.005$	$k_a = 0.1$
ν_{moves}	4.7015	4.8150	3.2110
$\bar{\Lambda}$	0.6667	0.6593	0.6502
\bar{C}_1	0.8000	0.8439	0.9000
\bar{C}_2	1.0000	0.9561	0.9000
d^+	21.1180	2.1680	1
d_1^-	13.5440	1.0920	1
d_2^-	28.6920	3.2440	1
ψ	0.9998	1	1

Table 4: Complete case

	$k_a = 0.001$	$k_a = 0.005$	$k_a = 0.1$
ν_{moves}	4.1580	4.5907	2.5242
$\bar{\Lambda}$	0.6667	0.6595	0.6511
\bar{C}_1	0.8000	0.8430	0.8933
\bar{C}_2	1.0000	0.9570	0.9067
d^+	6.3160	1.7000	1.0680
d_1^-	2.8720	1.0480	1.0240
d_2^-	9.7600	2.3520	1.1120
ψ	0.9998	0.9999	0.9986

Table 5: Regular case

resources in \mathcal{X}_2 are fully used, while for $k_a = 0.1$ resources in \mathcal{X}_1 and \mathcal{X}_2 are almost equally congested. Indeed in this last case, at least when the graph is complete, the aggregated solutions W^σ are optimal and they are indeed reached by the algorithm. Notice that the average degrees are already very small at the intermediate level $k_a = 0.005$: in this case the solution found by the algorithm is close to one where each unit uses just one resource. Clearly, the tuning of the parameters (in particular, the aggregation term k_a) will be important in real world applications: it will allow, depending on the specific needs of the working case, to trade-off reliability of the mostly used resources versus aggregation. Finally, notice how the fact of working with a preassigned topology does not change very much the behavior of the algorithm.

We conclude this section with an example where we also incorporate the feature that resources may be on or off. In our discrete time model we assume that every time each unit x is on with probability p_x (independently from the others). Specifically, we assume that \mathcal{X} is split into two subfamilies \mathcal{X}_1 and \mathcal{X}_2 such that $\lambda_x = \lambda^i$ and $p_x = p^i$ if $x \in \mathcal{X}_i$. Moreover, we assume that $\lambda^i = mp^i$ for $i = 1, 2$ and some proportionality constant m . This makes sense: the reliability is assumed to be proportional to the probability of finding the resource in on state.

Example 12. Assume that $n = 50$, $|\mathcal{X}_1| = |\mathcal{X}_2| = 25$, and that the underlying graph is complete. Parameters are set to the values: $\alpha = 45$, $\beta = 50$, $p^1 = 0.5$, $p^2 = 0.8$, $m = 5$, $k_a = 0, 1$. Here, we fix the time horizon $T = 50 * \sum_x \alpha_x$.

We compute the parameters ψ , ν_{moves} , d^+ , d^- , $\bar{\Lambda}$, and \bar{C}^i averaged over 10 running of the algorithm.

ν_{moves}	3.1080
$\bar{\Lambda}$	0.6620
\bar{C}_1	0.8278
\bar{C}_2	0.9722
d^+	1.4860
d_1^-	0.9320
d_2^-	2.0400
ψ	0.9484

Table 6: Values considering on and off resources

As shown in the table, even if the time horizon is higher than in previous examples, the average number of moves per each atom is of similar order due to the fact that when units are in off mode they can neither allocate or accept allocation. For the rest, the other parameters show that the algorithm, though more slowly, has a behavior similar to the case when units are always on.

5 Conclusions

In this paper, we have analyzed an allocation algorithm that allows a network of units (computers) to use the memory space available in their neighbors to store a back up of their data. The interest for such an algorithm comes from the goal of creating a peer-to-peer decentralized cloud storage model to be thought as an alternative to classical centralized cloud storage services.

The proposed algorithm is based on evolutionary game theory. The main theoretical contribution has been to prove convergence and to give an efficient characterization of the asymptotic in terms of an explicit invariant probability distribution picked on optimal Nash equilibria. We have also carried on a number of simulations showing the good scalability properties of the algorithm, its reduced complexity and illustrating the nice features of the final allocation state.

A number of important issues remain open and are left for future research. Among these: (i) find explicit bounds on the convergence time of the algorithm; (ii) study the structure of the Nash equilibria for general topologies; (iii) extend to the case when multiple back up copies are needed to be stored with the further security constraint of using different resources for the different copies.

Acknowledgment

We acknowledge that this work has been done while Barbara Franci was a PhD Student sponsored by a TIM/Telecom Italia grant.

References

- [1] Fagnani, F., Franci, B., Grasso, E. (2016) "A game theoretic approach to a peer-to-peer cloud storage model", MTNS 2016.
- [2] Han, Z., Niyato, D., Saad, W., Ba?ar, T., Hjorungnes, A. (2011) "Game Theory in Wireless and Communication Networks: Theory, Models, and Applications". Cambridge Univ. Press .
- [3] Jalaparti, V., Nguyen, G., Gupta, I., Caesar, M. (2010). "Cloud Resource Allocation Games". Technical Report, University of Illinois. <http://hdl.handle.net/2142/17427>
- [4] Lasaulce, S., Tembine, H (2011) "Game theory and learning for wireless networks: fundamentals and applications". Academic Press
- [5] Marden, J. R., Arslan, G., Shamma, J. S. (2009) "Cooperative Control and Potential Games" IEEE Transactions on Systems, Man and Cybernetics. Part B (Cybernetics), 39(1), 393-1407.
- [6] Marden, J. R., Shamma, J. S. (2014) "Game Theory and Distributed Control," Handbook of Game Theory Vol. 4, edited by Peyton Young and Shmuel Zamir, Elsevier Science.
- [7] Rosenthal, R. W. (1973). "A class of games possessing pure-strategy Nash equilibria". International Journal of Game Theory, 2(1), 65-67.

- [8] Tekin, C., Liu, M., Southwell, R., Huang, J., Ahmad, S. H. A. (2012). "Atomic congestion games on graphs and their applications in networking". *IEEE/ACM Transactions on Networking*, 20(5), 1541-1552.